

# 実行例: シクロヘキサンの2構造間の鞍点計算

2013 年 6 月 26 日

## 目 次

<b>1</b>	<b>はじめに</b>	<b>2</b>
<b>2</b>	<b>実行手順</b>	<b>2</b>
2.1	2つの安定状態の構造最適化 . . . . .	2
2.2	diffstr による 2 構造間の 3N 次元変位ベクトルの計算 . . . . .	2
2.3	iplstr による初期配置の作成 . . . . .	3
2.4	入力ファイルの作成 . . . . .	3
2.5	実行 . . . . .	4
<b>3</b>	<b>解説</b>	<b>5</b>
3.1	Hyper Plane Constraint 法 . . . . .	5
3.2	Force Inversion 法 . . . . .	6
3.3	共通 . . . . .	6

## 1 はじめに

ここでは活性化障壁の計算例として、シクロヘキサンの2つの準安定構造 (舟形構造と椅子型構造) の間の鞍点 (エネルギーと原子配置) を、xTAPP に実装されている超平面制限法 (Hyper Plane Constraint 法、以下 HPC 法と呼ぶ) または Force Inversion 法 (以下 FI 法と呼ぶ) によって求める方法を解説する。この配布パッケージには入力例と出力例として `sample/example-c6h12` が同梱されているので参照のこと。

通常の構造最適化のための入力データ作成はマスターしているものと想定し (`sample/tutorial` 参照)、ここでは活性化障壁の計算で特に必要になる部分だけを解説する。

## 2 実行手順

HPC 法と FI 法は活性化障壁を求める方法としてはかなり primitive な方法なので、詳しい解説の3節を良く読んで注意して使用すること。本節ではまず実際の実行手順について説明する。HPC 法と FI 法での違いを特に説明していない部分は、2つの方法に共通の手順である。

### 2.1 2つの安定状態の構造最適化

まず始状態と終状態になるべき2つの (準) 安定構造を求める。それぞれ通常の構造最適化を行えば良い。ただしここで、原子配置を記述する原子の順番には注意を払うこと。通常の個々の計算では、原子配置さえ正しければそれを記述する順番は何でも良いが、ここでは始状態と終状態の原子の記述の順番が一致 (対応) していなければならない。順番が一致していないと、変位しないはずの原子でも、別の位置の同種原子と入れ替わる、というような意図と違う遷移の計算になってしまう。

`c6h12-chair.cg` と `c6h12-ship.cg` はそれぞれ椅子型と舟形を初期状態として構造最適化を行なう入力ファイルである。ちなみにこれらは既に収束した座標を用いているので、`cgmprpt` を実行すると SCF を1回行なっただけで収束条件を満たして終了する。

結果として出力された99番ファイルにそれぞれ `chair.str`、`ship.str` という名前を付けるものとする。

### 2.2 diffstr による2構造間の3N次元変位ベクトルの計算

原子数が  $N$  であれば一つの状態は  $3N$  次元のベクトルで表現される。 $3N$  次元での始状態から終状態への変位ベクトルを付属の `diffstr` というプログラムを用いて求める。

```
% diffstr chair.str ship.str > vnormal.dat
```

で `vnormal.dat` に  $3N$  次元の変位ベクトルデータが出力される。このベクトルの方向を法線方向と呼ぶことにする。

なお、ユニットセルの境界をまたぐ様な変位がある場合にも、最短の変位となる様に自動的に計算される。

## 2.3 iplstr による初期配置の作成

HPC 法は、始状態から終状態への直線上の複数の地点を選び、それぞれを初期状態として、その直線に垂直な超平面内限定の原子配置最適化を行ない、反応経路を求めるものである。(図 1(a) 参照。) それぞれの入力データを作成するために、付属のプログラム iplstr を用いて 2 点間の中間の原子配置を求める。始点を 0、終点を 1 として、中間の位置を 0 から 1 の間の数字で指定する。

```
% iplstr chair.str ship.str 0.3 > str03.dat
```

で 0.3 の地点の原子配置のデータが str03.dat に出力される。

HPC 法では複数の地点で計算を行なう必要があるので、例えば添付の例のように 0.1 刻みで 0.1 から 0.9 までの 9 個のデータを作成する。

FI 法は鞍点に近いどこか 1 点から計算を開始し、法線方向にはエネルギーが上昇する方向に、それ以外の方向 (超平面内方向) にはエネルギーが低下する方向に動かして構造最適化を行なう方法である。このため、上記の iplstr を使って始状態と終状態の中間の 1 点を求めるというのが FI 法のための初期構造を作る一つの方法である。できるだけ鞍点に近い初期状態から始めた方が正しく収束する可能性も高まるので、HPC 法である程度鞍点に近い地点を求めてから FI 法を行なうという手順が推奨される。

なお、diffstr と同様、iplstr もユニットセルの境界をまたぐ様な変位に対しては、最短の変位となる様に自動的に計算される。

## 2.4 入力ファイルの作成

1. まず入力ファイルの # atom data セクションに、先ほど iplstr で求めた原子座標 (上記の例では str03.dat をそのまま) をコピーする。これは通常の構造最適化や SCF だけの計算の時と同様である。なお、対称性を原子構造に仮定する場合には以下のことに注意すべきである。

- 始状態と終状態が持つ対称性を鞍点の状態はもっていない可能性があること。
- HPC 法や FI 法で用いる 3N 次元法線ベクトルは仮定する対称性と矛盾してはならないこと。矛盾すると計算が途中で停止する。

2. 次に入力ファイルの # struct\_opt data のセクションに

```
mode = HYPER_PLANE,      ! HPC 法の場合  
mode = FORCE_INVERSION,  ! FI 法の場合
```

という記述のどちらかを入れる。

3. 次に # str\_opt\_constr data のセクションの tatom の入力の上に空行を 1 行開けてから、diffstr で求めた変位ベクトルデータ (上記の例では vnormal.dat を) コピーする。このセクション全体の例を以下に示す。

```
# str_opt_constr data  
4  
0.0 0.0 0.0  
0.0 0.0 0.0  
0.0 0.0 0.0
```

```

1.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0
1.0 0.0 0.0
0.0 1.0 0.0
0.0 0.0 0.0
3
2      1
1
3      1
13
4      1
8

```

```

0.000000000000 0.000000000000 0.000000000000 ! 1
0.000825711284 -0.000411964347 0.001622026935 ! 1
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000007990475 0.000370158125 0.000188542435 ! 1
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.001301166977 0.001207801348 0.000000000000 ! 1
0.000000000000 0.000000000000 0.000000000000 ! 2
-0.015620112051 -0.009461326538 -0.151140799680 ! 1
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.001892101951 0.000000000000 0.000000000000 ! 1
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2
0.000000000000 0.000000000000 0.000000000000 ! 2

```

最後の連続した 18 行がここで追加されたデータであり、これらが HPC 法や FI 法で用いる 3N 次元法線ベクトルである。なお、ここで 3.3 節の説明にしたがって、炭素原子以外の変位ベクトルは 0 に手でセットしている。

sample/example-c6h12 の中の 0.10 ~ 0.90 には HPC 法の入力データ例が、fi には FI 法の入力データ例が、PS にはこの計算で使う擬ポテンシャルが収められている。

## 2.5 実行

作成した入力ファイルを用いて cgmprpt を実行する。HPC 法の場合はそれを複数の地点に対して行なう。得られたエネルギーを内挿して得られるピーク地点が鞍点である。鞍点付近の分割を細かくすればより高い精度で鞍点が求められる。FI 法の場合は 1 回実行するだけだが、鞍点にある程度近い地点から計算を始めないと正しく収束しない場合も多い。

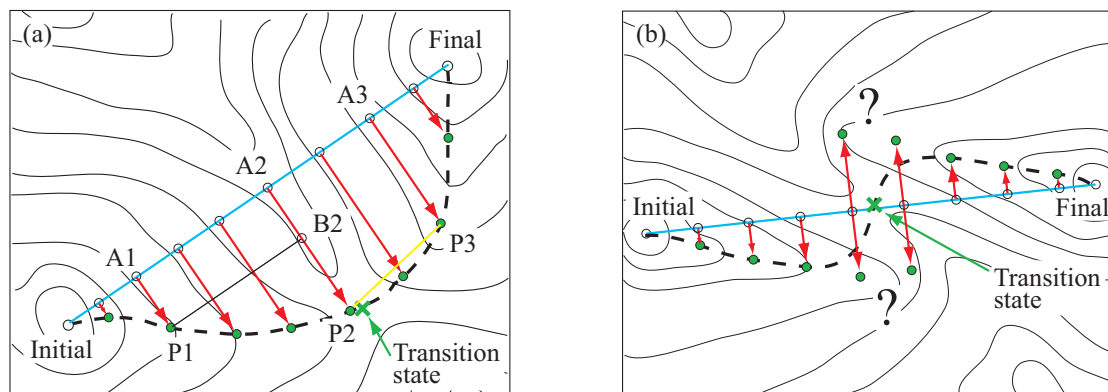


図 1: (a) 反応経路探索の例。初状態 (Initial) と終状態 (Final) を結ぶ線に適当に (この場合は 9 等分) を取り、それらの点から直線に垂直な超平面内でエネルギー最適化を行う。次は、反応経路の鞍点を含むと考えられる点の間を適当に分割して同様に反応経路を探索する。(b) 反応経路探索がうまくいかない例。初状態 (Initial) と終状態 (Final) を結ぶ線上から計算を始めた場合、線上の点でのポテンシャル面の傾きによって、どちらの谷に落ちるか分からない。

### 3 解説

#### 3.1 Hyper Plane Constraint 法

HPC 法の利点は、アルゴリズムが簡単なこと、鞍点を直接求める方法に比べて途中経路がわかるので、妥当性の判断がしやすいこと等がある。一方で、複雑な構造のポテンシャルの場合には鞍点が見付からない (図 1(b) 参照) とか、経路を計算する分だけ、計算時間が要求される、オートマチックには行かない等の欠点もある。

図 1(a) に 3N 次元空間内に存在する始状態 (Initial) と終状態 (Final) 及びそれらを囲むエネルギーポテンシャル面の等高線を 2 次元で模式的に表す。Initial と Final を結ぶ直線 (法線と呼ぶことにする) を適当に分割してそれらの点 (例えば A1-A3) を初期値として法線ベクトルに垂直な超平面内でエネルギー最適化を行うことにより、反応経路上の点 P1-P3 が得られる。

得られた反応経路上の点から、鞍点を含むと思われる区間を更に細分化して、反応経路上の点を求めることで、より精度良く鞍点を求めることができる。ただし、細分化して再計算をする際には、法線を変更してはならない。例えば、図 1(a) の P2-P3 間の黄色直線を新たに反応座標としたい。この場合は一見うまくいきそうに見えるが、実際には 3N-1 次元空間の最適化を行うので、図 1(a) のような 2 次元ポテンシャル面では想像できない経路を通して Initial か Final に近い点に落ちていってしまう可能性がある。従って、法線ベクトルは一旦 Initial と Final を決めたら途中で変えてはいけない。

- 拡散経路が長い場合は、途中で準安定状態をいくつか通ることが考えられる。その場合はまず、Initial と Final の間をザックリと分割して、超平面上の最小化を行う。すると、原子にかかる生のフォース (射影したフォースではなく) が十分小さくなる場所がある。その場合はその構造の近くに準安定構造があることが考えられるので、constraint をはずして構造を最適化して準安定構造を探す。準安定構造が見つかったら、拡散経路探索のパスを Initial → 準安定構造、準安定構造 → Final という風に分割して探索すると良い。
- 図 1(b) のような場合は探索がうまく行かない。反応座標 vs エネルギーの図を描くと、鞍点と思われる点の間が他の点に比べて明らかに広い場合がある。この場合は図 1(b) のようなポテンシャル面を探索している可能性がある。これは、鞍点の前後で図のように谷を乗り

換えるので、反応座標としては大きな距離を飛んでしまうことになるからである。この場合は、いくら鞍点の周りを細分化しても、飛びはなくなる。

### 3.2 Force Inversion 法

Force Inversion(FI) 法は鞍点に向かって原子配置を収束させて行く方法である。HPC 法では、原子配置最適化に使用する力ベクトルから法線方向成分を除いたが、FI 法では、力ベクトルから法線方向成分の 2 倍を引いたものを力ベクトルとして採用する。つまり

$$(1 - 2\mathbf{v}_n^t \mathbf{v}_n^t \mathbf{A} \mathbf{A}) \mathbf{f} \quad (1)$$

を力  $\mathbf{f}$  と考えて最適化を行うことになる。このようにして最適化を行うと法線ベクトル方向成分はエネルギーの上昇する方向へ変位し、直交する成分はエネルギーが降下する方向へ変位して行くわけであるから、谷を遡って行くような感じになり、最終的に上記の力ベクトルが 0 になれば、鞍点に到達したことになる。

- ポテンシャル面の性質が良い場合には通常の共役勾配法で良く収束するが、そうでなくうまく収束しない場合もある。安全性を高めるためには # struct\_opt data のセクションに

```
refresh_cycle = 1,
```

と設定する (つまり Steepest Descent 法を使用する) と良い。

- 電子状態が交差する場合は、FI 法ではうまく行かない。

### 3.3 共通

- 炭素と水素からなる有機分子などの場合、構造は主に炭素原子の位置で決定され、水素は炭素に結合しているだけ、と考えられる場合がある。その場合、超平面への制限は炭素原子だけに限定し、水素には制限を加えない方が鞍点探索がうまく行くことがある。これを実行するためには、diffstr で計算した変位ベクトルを手で編集して、水素原子のベクトルを 0 にしてしまえば良い。これは法線方向を水素への制限が無くなる方向に手動で変更する、という事を意味する。ここで挙げたシクロヘキサンの実行例では、そのような手動の変更を行なった後の変位ベクトルが使用されている。水素以外でも、最適化の可動範囲を制限したくない原子がある時はこの方法を使えば良い。